Proceedings of the RaPID-4 @LREC 2022, pages 62–70
Marseille, 25 June 2022

Speech Data Augmentation for Improving Phoneme Transcriptions of Aphasic Speech using wav2vec 2.0 for the PSST Challenge

Birger Moëll*, Jim O'Regan*, Shivam Mehta, Ambika Kirkland, Harm Lameris, Joakim Gustafsson, Jonas Beskow
Division of Speech Music and Hearing, KTH Royal Institute of Technology
{bmoell, joregan, smehta, kirkland, lameris, jkgu, beskow}@kth.se

Abstract

As part of the PSST challenge, we explore how data augmentations, data sources, and model size affect phoneme transcription accuracy on speech produced by individuals with aphasia. We evaluate model performance in terms of feature error rate (FER) and phoneme error rate (PER). We find that data augmentations techniques, such as pitch shift, improve model performance. Additionally, increasing the size of the model decreases FER and PER. Our experiments also show that adding manually-transcribed speech from non-aphasic speakers (TIMIT) improves performance when Room Impulse Response is used to augment the data. The best performing model combines aphasic and non-aphasic data and has a 21.0% PER and a 9.2% FER, a relative improvement of 9.8% compared to the baseline model on the primary outcome measurement. We show that data augmentation, larger model size, and additional non-aphasic data sources can be helpful in improving automatic phoneme recognition models for people with aphasia.

Keywords: aphasia, phoneme transcription, wav2vec 2.0, speech, phonemes, data augmentation, speech data augmentation

1. Introduction

Aphasia is a dysfunction of the ability to understand or produce language caused by damage to brain regions used for speech (Damasio, 1992). A common, broad distinction made in classifying different forms of aphasia is between fluent and non-fluent aphasia (Feyereisen et al., 1991). While those with fluent aphasias, such as Wernicke's aphasia, are typically able to produce syntactically and phonetically well-formed utterances, non-fluent aphasias such as Broca's aphasia and transcortical motor aphasia are characterized by difficulties in selecting and ordering phonemes and forming syntactically complex utterances. However, while most clinicians use fluency classifications in their diagnoses, the distinction is not well-defined (Gordon, 1998), and there is evidence that even so-called fluent aphasias involve errors in phoneme production (Blumstein et al., 1980; Kurowski and Blumstein, 2016; Vijayan and Gandour, 1995; Holloman and Drummond, 1991), possibly as a result of impaired acoustic-phonological control (Robson et al., 2012).

This phenomenon of inserting, deleting or substituting phonemes is known as phonemic paraphasia. Examples of this based on a related yet distinct clinical population with similar symptomatology include *lat* for *bat*, or *dake* for *drake*. The errors are concentrated on nouns and verbs, and occur evenly on vowels, single consonants, and consonant clusters (Dalton et al., 2018). For consonants, erroneous productions most commonly differ from the target phoneme by a single phonetic feature, though errors containing multiple phonetic feature differences occur as well. Substitution errors occur more commonly than insertion or deletion errors. These unintended phoneme substitutions are believed to be caused by a cascading activation of a target and a competitor phonetic segment with a speech output showing properties of both the target and competitor phonemes (Kurowski and Blumstein, 2016).

Several studies have shown that reliable phonemic annotation can be beneficial in the diagnosis of aphasia, and its distinction from acquired apraxia of speech (Cunningham et al., 2016), with phoneme distortion error rates being lower for patients with phonemic paraphasia. Error profiles can also be used as an indicator for the possibility of remediation of these phonological errors, as individuals displaying phonological errors display less improvement than individuals displaying motoric errors on a repetition training task (Buchwald et al., 2017). Finally, phonemic transcriptions are an important component in the development of individualized intervention plans for patients with aphasia (Abel et al., 2007). The ability to automatically transcribe the speech of aphasic patients would allow for a richer profile of data for each individual with less burden on the clinician. Automatic speech recognition (ASR) has been proposed as a valuable tool for developing effective speech therapy interventions (Jamal et al., 2017), but achieving robust, high-accuracy ASR for aphasic speech remains a challenge. Conventional ASR systems struggle with aphasic speech because of the irregularities of aphasic speech, so aphasiatic ASR systems needs to be trained specifically on aphasic speech.

In this paper we explore how speech data augmentations, data sources and model parameters can be optimized to create a robust, high accuracy phoneme transcription model for aphasic speech. We hope to give

* Equal contribution.

the reader an intuition about the steps involved in the creation of such a model with the aim of describing our work in such detail that it can be easily reproduced.

1.1. Phoneme Feature Vectors

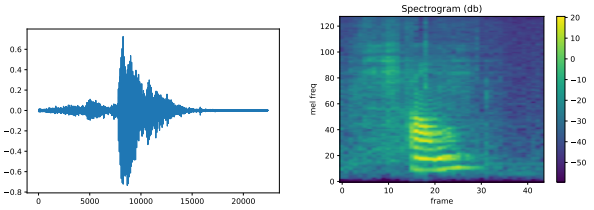
The goal of the Post-Stroke Speech Transcription (PSST) challenge is to create accurate automatic transcriptions of phonemes produced by speakers with aphasia. To this end, we use phonemic feature vectors in order to more precisely quantify the degree to which a produced phoneme differs from a target phoneme. A phoneme feature vector maps phonemes to their articulatory correlates (Chomsky and Halle, 1968). The features correspond to aspects such as vocal tract cavity configurations, place and manner of articulation, glottal states of sounds, and tongue body positions. A value of [+] for a given feature indicates that the feature is present, [-] indicates that it is absent, and [0] indicates that a phoneme is unmarked with respect to that feature (i.e., the feature is not relevant for defining the phoneme). For example, the consonant /f/ is [- voice] while the consonant /v/ is [+ voice]. Feature error rate (FER) allows for a more fine-grained analysis of errors in aphasic speech, penalizing errors that sound more similar to the target less severely, in contrast to phoneme error rate (PER), which does not indicate how dissimilar a produced phoneme is from a target phoneme and treats all incorrect productions equally.

1.2. Models for Aphasia Prediction

Recently, Self Supervised Learning (SSL) has attracted a lot of interest in all data modalities because of the high cost of annotation of data; models like BERT (Devlin et al., 2019), SimCLR (Chen et al., 2020b) have shown the ability to learn in a self supervised setting, either by predicting the next token or by contrastive learning. SSL is especially useful in the audio modality, mainly because of the presence of an abundance of unannotated audio data on the internet. With recent advances in deep learning, architectures like HuBERT (Hsu et al., 2021) and wav2vec 2.0 (Baevski et al., 2020) have shown results on par with supervised learning methods while reducing the overhead of gathering annotated data. In this work, we explore wav2vec 2.0 Base and Large models with various data augmentation methodologies to transfer the speech recognition knowledge of the pre-trained model to speech generated by a person with aphasia.

1.3. Data Augmentation

Many deep learning pipelines incorporate data augmentation as an important technique to achieve state-of-the-art results (Chen et al., 2020a). It is known to improve generalisation and learn translation invariance, which is useful for the models to learn the underlying structure of data instead of specific aspects of the training samples, resulting in better performance (Worrall et al., 2017). It has shown-state-of-the-art results in different modalities such as images (Krizhevsky et al., 2012) and text (Feng et al., 2021). Data augmentation has also been applied successfully in the audio modality, resulting in major improvement in speech classification and speech recognition (Tak et al., 2022). In this paper we augment the audio data in the waveform domain, giving us more training samples while maintaining the i.i.d assumption of the empirical data samples.



(a) Waveform (b) Mel Spectrogram

Figure 1: A sample from PSST dataset

2. Data

2.1. Datasets

In our experiments we explored how combining and augmenting data could help improve our predictions. We explored how training on the PSST, TIMIT, and Common Voice datasets affected model performance. Data statistics are summarised in Table 1.

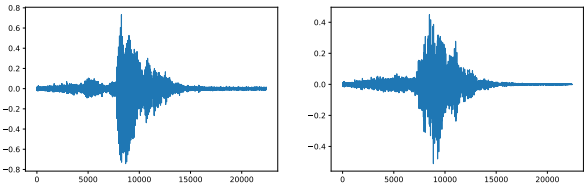
2.1.1. PSST

The PSST challenge dataset consists of a subset of the AphasiaBank data (MacWhinney et al., 2011) annotated with manually transcribed phonemes and made available through the python package psst-data (Gale et al., 2022). The data consists of 2298 utterances in the training dataset, 341 utterances in the validation dataset and 652 utterances in the test dataset. A sample from the dataset is visualised in Figure 1. Speakers with several different types of aphasia, as categorized by the Western Aphasia Battery (WAB) (Risser and Spreen, 1985), were represented in the training dataset. Of the 73 speakers, 26 had anomic aphasia, 18 had conduction aphasia, 18 had Broca's aphasia, 8 had Wernicke's aphasia, 2 had transcortical motor aphasia, and one speaker was classified as not aphasic based on their WAB results.

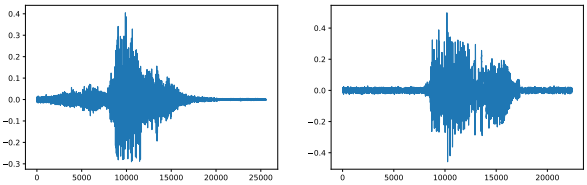
2.1.2. TIMIT

TIMIT (Garofolo et al., 1993) is the most commonly used dataset for phoneme recognition, as it is one of the few datasets available with phoneme labels (Lopes and Perdigao, 2011). Although TIMIT, like the PSST data, uses a phoneme set based on ARPAbet, it is based on a revised version. While, for the most part, there is a simple mapping to the version of ARPAbet used in the PSST data, there are three items[1] that do not map exactly. To avoid introducing imprecision into the training data, we elected to choose only segments that did

[1] dx (flap), nx (nasal flap), and q (glottal stop).

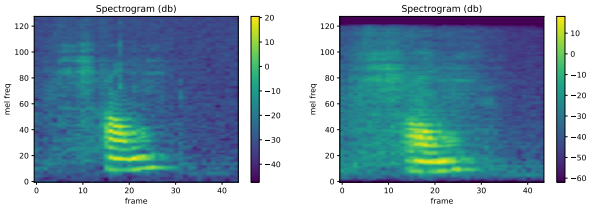


(a) Gaussian noise (b) Pitch shift

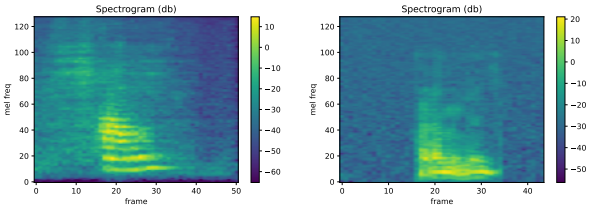


(c) Time stretch (d) Voice Conversion

Figure 2: Effect of data augmentation in the waveform



(a) Gaussian noise (b) Pitch shift



(c) Time stretch (d) Voice Conversion

Figure 3: Effect of data augmentation in mel spectrogram

not include these three items; as the number of segments was quite low, we also drew from the test set. In total, 1414 segments were used (1016 from train, 398 from test)[2].

2.1.3. Common Voice

Common Voice is a crowdsourced dataset of speakers of different languages. We used a subset of the English Common Voice with automatically added ARPA-phonemes using the open source python g2p package.[3]

Dataset	Number of segments	Manually transcribed	Audio (mins.)
PSST	2298	Yes	166
TIMIT	1414	Yes	64
Common Voice	15777	No	1559

Table 1: Dataset overview

2.2. Data Augmentation

We used the open source audiomentations library[4] to augment the PSST data as well as other datasets used in training. In our data augmentation we strove both to augment the available samples of the PSST Dataset to increase their number still keeping the dataset balanced and similar to the original PSST dataset, and to induce the noisy artefacts of PSST dataset into TIMIT. Figure 2 shows the effect of different type of waveform augmentation on the waveform of a sample audio and Figure 3 shows the effect of the same sample in the mel spectrogram domain.

[2]A list of IDs used, along with a fine-tuned model, is included in `https://huggingface.co/jimregan/psst-partial-timit`.

[3]`https://pypi.org/project/g2p-en/`

[4]`https://github.com/iver56/audiomentations`

2.2.1. Gaussian noise

Though seemingly paradoxical, adding noise to the data acts as regularization and improves generalization (Bishop, 1995). Gaussian noise is a common data augmentation: at each time a datapoint is exposed to the model a stochastic noise sampled from a standard Gaussian $\mathcal{N}(0, 1)$ is added to it making it different. Noise amplitude σ is a hyperparameter uniformly distributed over the range $\sigma \sim U(0.005, 0.015)$. The newly generated samples after augmentation can be represented as:

$$x(t) = x(t) + \sigma \times \mathcal{N}(0, 1)$$

The effect of this data augmentation is visible in Figure 2a for the waveform and Figure 3a for the mel spectrogram.

2.2.2. Time stretch

Time stretch is a data augmentation where the audio file is either sped up or slowed down without affecting the pitch. In theory this would improve generalization by making the model more independent of speaking rate. Generally γ is the stretch factor, if $\gamma > 1$ then the speed of the audio is increased and if $\gamma < 1$ then the speed of the audio is reduced. The stretch factor is uniformly distributed over $\gamma \sim U(0.8, 1.25)$. The augmentation results of this transformation on the original waveform can be seen in Figure 2c for the waveform and in Figure 3c for the mel spectrogram.

2.2.3. Pitch shift

We use pitch shift to vary the pitch of the signal. This improves generalization by helping learn a latent space independent of fundamental frequency. Pitch shift modifies the pitch of the audio sample either by raising or lowering the pitch while keeping the duration of the audio unchanged (Salamon and Bello, 2017). It is, in some ways, an inverse of the time stretch augmentation. We shifted individual samples by n semitones without

changing the tempo where $n \sim U(-4, 4)$. Figure 2b and Figure 3b visualise the effect of this transformation in both the waveform and the mel spectrogram.

2.2.4. Voice Conversion

We used the official open source implementation[5] of (Chou et al., 2019) to do one-shot voice conversion of audio files to improve the variability in data and make the data more speaker independent. They use a Variational Auto Encoder (Kingma and Welling, 2013) as a generative model with two encoders, where one is a context encoder while the other is a speaker encoder, with the use of Instance Normalization (IN) (Ulyanov et al., 2016) and Adaptive Instance Normalization (AdaIN) (Huang and Belongie, 2017) they synthesise the text conditioned on the target speaker representation. In our experiments, for all of the audio files of each speaker, the target audio file was chosen at random from all other speakers and was augmented to their speaker characteristics. This gave us varied samples of the same utterance but with different speaker characteristics. Since this method looks for voiced segments in the mel domain, output from these are shorter than others, but for visualisation only we have padded it with Gaussian noise to make it visually similar to 2.2.1. This padding was not used while training the model. The effect of this can be seen in both the waveform Figure 2d and the mel spectrogram Figure 3d.

2.2.5. Room Impulse Response

Room Impulse Response (RIR) augmentation is a technique for simulating room acoustics (Habets, 2006) by adding artificial reverberation. Given the variability in the acoustics of the recording environments of the AphasiaBank dataset, RIR might make it possible to bridge the acoustic gap when using othe datasets. The audiomentations library uses a wave-based technique, where recordings with the reverberance qualities of a particular room have been isolated and applied to the input using a convolution operation. We used two sets of publicly available impulse responses: EchoThief[6] and the MIT McDermott dataset[7], from which a recording is selected at random for application to the utterance.

2.3. Data Processing

All data was processed to work with the fairseq (Ott et al., 2019) framework in order to standardize the training process.

2.4. Model Architecture

For training we chose to fine-tune wav2vec 2.0. We experimented with Base and Large model. Although later models like Large (LV-60k) has shown better results we wanted to focus our experiments on data augmentations and how they affect model performance.

wav2vec 2.0 Model	**Base**	**Large**
Transformer blocks	12	24
Attention heads	8	16
Model dimension	768	1024
Inner dimension	3072	4096

Table 2: wav2vec 2.0 model variants and hyperparameters.

2.4.1. Wav2vec 2.0

wav2vec 2.0 (W2V2) is an architecture proposed in Baevski et al. (2020) that uses self-supervision in the audio domain to create audio vectors that can be used in training. The model consists of a multi-layer convolution feature encoder that takes as input raw audio and outputs latent speech representations. These latent representations are then fed to a Transformer to build representations that has the ability to capture information from the whole length of the sequence. This is done through a masking function in the audio domain. For our training, we chose to focus on the wav2vec 2.0 base model and the wav2vec 2.0 large model, to make a comparison of how model size affects and interacts with other techniques used while training. The model hyperparameters are mentioned in Table 2.

2.4.2. Fine-tuning

Pre-trained base models are fine-tuned for phoneme (and speech) recognition by adding a linear projection on top of the model, used to classify into the number of tokens found in the phoneme vocabulary (42).

2.4.3. Language Model

Language modelling refers to the use of various statistical and probabilistic methods to estimate the probability of a sequence of words. Formally, we can formulate the task of language modelling as

$$\begin{aligned} p(x_1, \ldots, x_t) &= p(x_1).p(x_2|x_1).\ldots.p(x_t|x_{<t}) \\ &= \sum_{i=1}^{i=t} p(x_i|x_{i-1}, \ldots, x_1) \end{aligned}$$

where x_i are the tokens in a sentence.

2.5. Evaluation

2.5.1. Phoneme Error Rate

Phoneme error rate is the number of phoneme errors (edits, insertions, and substitutions) divided by the number of phonemes in the reference transcript, calculated using the Levenshtein distance (Levenshtein, 1966).

[5] https://github.com/jjery2243542/adaptive_voice_conversion

[6] http://tulrich.com/recording/ir_capture/

[7] https://mcdermottlab.mit.edu/Reverb/IR_Survey.html

$$PER = 100 * \frac{\#Edits}{\#Phones}$$

2.5.2. Feature Error Rate

Feature error rate is the number of phoneme feature errors where phonemes which differ by fewer features are considered more correct. Transcribed phonemes are converted into phoneme feature vectors in order to calculate the feature error rate using the Levenshtein distance.

$$FER = 100 * \frac{\#Edits}{\#Features}$$

3. Experiment

In order to improve reproducibility we kept the hyperparameters constant using the same parameters as those used in the psst-baseline training. [8] We trained in a warm state manner with 4000 warm updates keeping learning rate at 5e-05 using the Adam optimizer to train the model.

Table 3 contains a summary of the best performing models.

3.1. Base Models

Two pre-trained wav2vec 2.0 models were used as base models for all experiments: "wav2vec 2.0 Base" and "wav2vec 2.0 Large" are the "No finetuning" versions of the models, as found in the fairseq GitHub repository[9].

3.2. PSST Augmentations

We augmented the PSST dataset with augmentations defined in Section 2.2. We used Gaussian noise as a data augmentation for the base model and pitch shift and time stretch independently as augmentations for two large models. There was a 50% probability of the data being augmented, with the augmented dataset doubling in size compared to the non-augmented data with on average 25% augmented data, 25% overlapped data and 50% consisting of the original data.

3.3. PSST with Augmented TIMIT

As speech recognition models can often be sensitive to differences in acoustic conditions; it is not automatically the case that additional data will lead to an improvement when there is a difference in recording conditions. Because of the mismatch of recording conditions between TIMIT, which was recorded in clean conditions, and the PSST data, which was not, we experimented with augmenting the TIMIT data alone, to attempt to artificially match the PSST data. As well as Gaussian noise, pitch shift, and time stretch, we also added RIR to match the dry, studio conditions of TIMIT to PSST.

3.4. Language Model

To explore the effect of the language model, we augmented the transcription data of the combined PSST and TIMIT datasets with the CMU Pronouncing Dictionary (CMUdict)[10], across configurations of 4-, 5-, and 6-gram models[11]. We used two versions of the PSST+TIMIT data: unmodified, and with silence tokens removed (and the spoken noise token, in the case of PSST); to emulate the silence between words with CMUdict, we used the unmodified entries, entries with a silence token added at the start, added at the end, and added at both start and end, with an additional "all silences" configuration which combined all configurations.

4. Results

The results of our experiments are summarised in Table 3 and Figure 4. While evaluating on the PSST validation dataset we found improved scores for several techniques.

While some training heuristics–such as adding an n-gram language model and using data augmentation such as Voice Cloning, Gaussian Noise and Time-stretch–had results comparable to the baseline trained on PSST dataset with wav2vec 2.0 (FER: 10.2, PER: 22.2), other configurations lead to improved results.

The wav2vec 2.0 large model trained on the PSST data had a relative improvement of 5.86% for PER (20.9 vs 22.2) and 3.92% for FER (9.8 vs 10.2).

The wav2vec 2.0 large model trained on the PSST data with pitch shift improved the scores by 4.5% for PER (21.2 vs 22.2) and 6.86% for FER (9.5 vs 10.2).

The wav2vec 2.0 large model trained on the PSST data with pitch shift + TIMIT improved the scores by 4.5% for PER (21.2 vs 22.2) and 7.3% for FER (9.7 vs 10.2).

The wav2vec 2.0 base model trained on the PSST data + TIMIT with RIR achieved the best score of the various combinations of augmentations described in section 3.3, improving the scores by 1.8% for PER (21.8 vs 22.2) and 5.88% for FER (9.6 vs 10.2).

The wav2vec 2.0 large model trained on the PSST data + TIMIT with RIR achieved the best overall score, improving the results by 5.41% for PER (21.0 vs 22.2) and 9.8% for FER (9.2 vs 10.2).

As part of our experiments we also reproduced the baseline model. Our reproduced baseline had lower scores than the PSST Baseline by 1.96% for PER (10.4 vs 10.2) and 4.05% for FER (23.1 vs 22.2). The difference could be caused by initial weight randomization. We choose to compare all our models to the original baseline model.

[8] https://github.com/PSST-Challenge/psstbaseline

[9] https://github.com/pytorch/fairseq/tree/main/examples/wav2vec/

[10] https://github.com/cmusphinx/cmudict

[11] 6 is the maximum number of n-grams supported by the default configuration of the language model library used by the PSST Challenge scripts.

Name	Data	Model	FER	PER
PSST Baseline	PSST	Base	10.2%	22.2%
Reproduced Baseline	PSST	Base	10.4%	23.1%
Common Voice	Common Voice phonemes	Base	61.8%	91.6%
Baseline + TIMIT RIR	PSST Partial TIMIT with RIR	Base	9.6%	21.8%
Gaussian Noise (DA)	PSST with Gaussian Noise	Base	9.9%	22.9%
W2V2 Large	PSST	Large	9.8%	**20.9%**
W2V2 Large Voice Clone	PSST + Voice clone	Large	10.3%	22.7%
W2V2 Large Time-Stretch	PSST Time Stretch	Large	10.0%	21.2%
W2V2 Large Pitch-Shift	PSST Pitch Shift	Large	9.5%	21.2%
W2V2 Pitch-Shift + TIMIT RIR	PSST Pitch Shift + TIMIT RIR	Large	9.7%	21.2%
W2V2 Large + TIMIT RIR	PSST + TIMIT RIR	Large	**9.2%**	21.0%

Table 3: Experimentation results with different combinations of model and augmentations

Furthermore, we evaluated training on Common Voice and TIMIT without PSST, finding that these models were not successful at aphasic phoneme recognition without fine-tuning on aphasic speech. We also continued fine-tuning Common Voice on PSST with poor results. The poor results on Common Voice could be related to the automatic phoneme transcriptions which might not have been comparable to manually transcribed phonemes.

Several models showed improvements in PER without improvements in FER. One hypothesis is that this is due to the manner of calculation of FER versus PER per phoneme, PER has a binary outcome whereas FER is averaged over 20 features hence leading to less variation in the score for FER.

4.1. Language Models

The best performing language model, 5-gram with silences removed from PSST and TIMIT, but with CMUdict data with silence tokens added at the end, achieved PER of 22.1%, compared with the baseline of PSST and nonaugmented TIMIT without a language model (PER 22.5%). No difference in FER was observed with any language model configuration. A plot of the results of this language model and a selection of the results from section 3.3 can be viewed in figure 4.

4.2. Model Availability

The models are available for download on Huggingface[12].

5. Discussion

In this paper, we looked at the challenges of the current Automatic Speech Recognition (ASR) techniques for the low-resource task of aphasic phoneme recognition, and devised heuristics for improving the phoneme transcriptions.

Training with a larger baseline model was one of the most straightforward ways to improve performance. In general, all the models trained with wav2vec 2.0 Large outperformed similar models trained with wav2vec 2.0 Base. This is in line with the current trend in deep learning, where larger self-supervised transformer models outperform the state of the art by keeping architecture similar while increasing model size. However, training on larger models has several drawbacks, one being increased training and inference time, another being the need for specialised GPUs that might be expensive to acquire or use. If computation is a bottleneck, it might be sensible to start by training a smaller model with different parameters and later train a larger model after good parameters have been found that improve performance.

Data augmentations on PSST was another technique that improved the performance. Pitch shift was the most useful augmentation technique when outside data sources were not used, with models using pitch shift showing good results especially on FER. Pitch shift transformation could be viewed as a transformation of

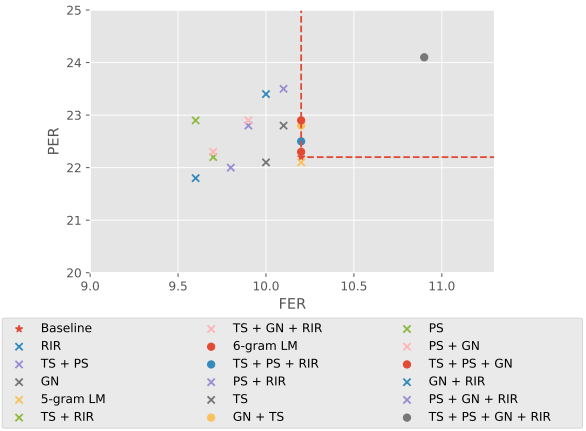


Figure 4: Sample results of TIMIT augmentation and language model experiments, using Gaussian Noise (GN), Time Stretch (TS), Pitch Shift (PS), and Room Impulse Response (RIR). LM results are on PSST/TIMIT with silences removed, augmented with CMUdict with appended silence tokens. Results to the left of the vertical line represent improvements in FER, while results below the horizontal line represent improvements in PER.

[12] https://huggingface.co/birgermoell

the vocal tract length and vocal fold of the speaker, which could help the model to generalise the difference between phonemic features and make the model more speaker independent. Given more time, experiences with pitch shift parameters might have the potential to improve accuracy further, in line with previous research (Salamon and Bello, 2017).

While working with data augmentation it is important that the underlying structure of the data is preserved, i.e., data augmentation should aim to help the model learn by augmenting features in the dataset, but not change the features so much that the underlying signal in the data gets corrupted. Voice cloning was an experiment where the data augmentation might have failed in this regard and the augmented samples had, in general, a lower pitch than the originals. When working with data augmentation, we believe that an inspection of the augmented data itself is a good first step in determining if the data will be useful for training. Here, common sense reasoning by a person knowledgeable in the field should suffice. If the data sounds reasonable, it has the potential to be helpful for improving model performance. This might seem obvious, but in the paradigm of large training sets and large models we still want to emphasize the importance of keeping a human in the loop.

A limitation in our work is the small size of the PSST dataset and the modest improvements we made compared to the baseline. The small dataset size makes it harder to determine how well our models have generalised. When working with deep learning models it is always hard to determine how parameters interact and we think it is sensible to view this work as a way to understand data augmentation in the aphasic phoneme domain rather than seeing it as a recipe for achieving state of the art.

An interesting scientific question is: to what degree do aphasic phonemic speech models improve by training on different data sources consisting of non-aphasic speech?

We found that training a model only on Common Voice or TIMIT was not sufficient to get a working model. This shows that at least in our experiment some part of the data needs to be aphasic. Furthermore, we continued fine-tuning on PSST from the model trained on Common Voice with limited results. This might be because Common Voice was automatically transcribed, but it may be related to the order of training.

In our experiment we found that the best performing model trained on TIMIT + PSST is close in performance to the best performing model trained only on PSST data. Here, data augmentations on TIMIT using RIR to make the data sound similar to PSST clearly helped performance by bringing the datasets more into alignment.

In theory, a similarly performing model that is trained on both aphasic and non-aphasic speech is preferable, as it has the potential to generalise better. Since our best performing model uses both aphasic and non-aphasic speech, a fair conclusion is that non-aphasic speech prepared in the proper format is a data source augmentation worth exploring when working with aphasic data.

A well-functioning phonetic and feature error prediction model for aphasia appears a promising way forward in order to build automated electronic tools for aphasia recovery.

Improved understanding of aphasia through automated tools for testing might also help determine which individuals are most helped by specific interventions.

6. Conclusion

In conclusion, our paper has shown that data augmentation, larger model size and additional non-aphasic data sources can be helpful in improving automatic phoneme recognition models for people with aphasia.

7. Acknowledgements

The results of this work and the tools used will be made more widely accessible through the national infrastructure Språkbanken Tal under funding from the Swedish research Council (2017-00626). We would like to thank the 509 scientific community for their support of our work. This research was also supported by the Swedish Research Council project Connected (VR-2019-05003) and partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation and funded by Swedish Foundation of Strategic Research (SSF), project EACare under Grant No RIT15-0107.

8. Bibliographical References

Abel, S., Willmes, K., and Huber, W. (2007). Model-oriented naming therapy: Testing predictions of a connectionist model. *Aphasiology*, 21(5):411–447.

Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Bishop, C. M. (1995). Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 7(1):108–116, 01.

Blumstein, S. E., Cooper, W. E., Goodglass, H., Statlender, S., and Gottlieb, J. (1980). Production deficits in aphasia: A voice-onset time analysis. *Brain and language*, 9(2):153–170.

Buchwald, A., Gagnon, B., and Miozzo, M. (2017). Identification and remediation of phonological and motor errors in acquired sound production impairment. *Journal of Speech, Language, and Hearing Research*, 60(6S):1726–1738.

Chen, S., Dobriban, E., and Lee, J. (2020a). A group-theoretic framework for data augmentation. *Advances in Neural Information Processing Systems*, 33:21321–21333.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020b). A simple framework for contrastive learning of visual representations. In Hal Daumé III et al., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul.

Chomsky, N. and Halle, M. (1968). The sound pattern of english.

Chou, J.-C., chieh Yeh, C., and yi Lee, H. (2019). One-shot voice conversion by separating speaker and content representations with instance normalization. In *INTERSPEECH*.

Cunningham, K. T., Haley, K. L., and Jacks, A. (2016). Speech sound distortions in aphasia and apraxia of speech: Reliability and diagnostic significance. *Aphasiology*, 30(4):396–413.

Dalton, S. G. H., Shultz, C., Henry, M. L., Hillis, A. E., and Richardson, J. D. (2018). Describing phonological paraphasias in three variants of primary progressive aphasia. *American journal of speech-language pathology*, 27(1S):336–349.

Damasio, A. R. (1992). Aphasia. *New England Journal of Medicine*, 326(8):531–539.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online, August. Association for Computational Linguistics.

Feyereisen, P., Pillon, A., and Partz, M.-P. d. (1991). On the measures of fluency in the assessment of spontaneous speech production by aphasic subjects. *Aphasiology*, 5(1):1–21.

Gale, R., Fleegle, M., Bedrick, S., and Fergadiotis, G. (2022). Dataset and tools for the PSST Challenge on Post- Stroke Speech Transcription, March. Project funded by the National Institute on Deafness and Other Communication Disorders grant number R01DC015999-04S1.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Pallett, D. S., Dahlgren, N. L., Zue, V., and Fiscus, J. G. (1993). TIMIT Acoustic-Phonetic Continuous Speech Corpus. Type: dataset.

Gordon, J. K. (1998). The fluency dimension in aphasia. *Aphasiology*, 12(7-8):673–688.

Habets, E. A. (2006). Room impulse response generator. *Technische Universiteit Eindhoven, Tech. Rep*, 2(2.4):1.

Holloman, A. L. and Drummond, S. S. (1991). Perceptual and acoustical analyses of phonemic paraphasias in nonfluent and fluent dysphasia. *Journal of communication disorders*, 24(4):301–312.

Hsu, W.-N., Tsai, Y.-H. H., Bolte, B., Salakhutdinov, R., and Mohamed, A. (2021). Hubert: How much can a bad teacher benefit asr pre-training? In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6533–6537.

Huang, X. and Belongie, S. (2017). Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct.

Jamal, N., Shanta, S., Mahmud, F., and Sha'abani, M. (2017). Automatic speech recognition (asr) based approach for speech therapy of aphasic patients: A review. In *AIP Conference Proceedings*, volume 1883, page 020028. AIP Publishing LLC.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, et al., editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Kurowski, K. and Blumstein, S. E. (2016). Phonetic basis of phonemic paraphasias in aphasia: Evidence for cascading activation. *Cortex*, 75:193–203.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Lopes, C. and Perdigao, F. (2011). *Phoneme Recognition on the TIMIT Database*. IntechOpen, June.

MacWhinney, B., Fromm, D., Forbes, M., and Holland, A. (2011). Aphasiabank: Methods for studying discourse. *Aphasiology*, 25(11):1286–1307.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Risser, A. H. and Spreen, O. (1985). The western aphasia battery. *Journal of clinical and experimental neuropsychology*, 7(4):463–470.

Robson, H., Sage, K., and Ralph, M. A. L. (2012). Wernicke's aphasia reflects a combination of acoustic-phonological and semantic control deficits: a case-series comparison of wernicke's aphasia, semantic dementia and semantic aphasia. *Neuropsychologia*, 50(2):266–275.

Salamon, J. and Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal processing letters*, 24(3):279–283.

Tak, H., Todisco, M., Wang, X., Jung, J.-w., Yamagishi, J., and Evans, N. (2022). Automatic speaker verification spoofing and deepfake detection using

wav2vec 2.0 and data augmentation. In ISCA, editor, *Submitted to ODYSSEY 2022, The Speaker Language Recognition Workshop, June 28th-July 1st, 2022, Beijing, China*, Beijing. © ISCA. Personal use of this material is permitted. The definitive version of this paper was published in Submitted to ODYSSEY 2022, The Speaker Language Recognition Workshop, June 28th-July 1st, 2022, Beijing, China and is available at :.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization.

Vijayan, A. and Gandour, J. (1995). On the notion of a "subtle phonetic deficit" in fluent/posterior aphasia. *Brain and Language*, 48(1):106–119.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. (2017). Harmonic networks: Deep translation and rotation equivariance. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7168–7177, Los Alamitos, CA, USA, jul. IEEE Computer Society.

A. Experimental details

A.1. TIMIT augmentations

Table 4 contains the results of the augmentations using the 64 minutes of TIMIT (see subsection 2.1.2, above). The name of the augmentation in the table corresponds with the branch name of the git repository[13].

Augmentation	FER	PER
unaugmented	10.2%	22.5%
gaussian	10.0%	22.1%
pitchshift	9.6%	22.9%
rir	**9.6%**	**21.8%**
timestretch	10.1%	22.8%
gaussian-rir	10.0%	23.4%
pitchshift-gaussian	9.9%	22.9%
pitchshift-rir	9.9%	22.8%
timestretch-gaussian	10.2%	22.8%
timestretch-pitchshift	9.8%	22.0%
timestretch-rir	9.7%	22.2%
pitchshift-gaussian-rir	10.1%	23.5%
timestretch-gaussian-rir	9.7%	22.3%
timestretch-pitchshift-gaussian	10.2%	22.9%
timestretch-pitchshift-rir	10.2%	22.5%
timestretch-pitchshift-gaussian-rir	10.9%	24.1%

Table 4: Results of combining various augmentations of TIMIT with the unaugmented PSST data.

A.2. Language model experiments

Table 5 contains the results of all permutations of the experiments with language models (see subsection 2.4.3, above). The models are contained in the same git repository as the TIMIT augmentations; the README accompanying the repository contains a mapping of branches to the experiment.

	n-gram	FER	PER
Baseline + TIMIT	–	**10.2%**	22.5%
All silences	4	10.5%	23.0%
	5	10.5%	22.6%
	6	10.3%	22.3%
No silences	4	10.3%	22.6%
	5	**10.2%**	22.2%
	6	**10.2%**	22.4%
PSST and TIMIT without silence			
CMUdict-end	4	10.3%	22.6%
	5	**10.2%**	**22.1%**
	6	**10.2%**	22.3%
CMUdict-start	4	10.4%	22.6%
	5	10.3%	22.4%
	6	10.3%	22.3%
CMUdict-both	4	10.4%	22.7%
	5	10.4%	22.3%
	6	10.3%	22.3%
Unmodified PSST and TIMIT			
Unmodified CMUdict	4	10.3%	22.8%
	5	10.3%	22.4%
	6	**10.2%**	22.4%
CMUdict-end	4	10.3%	22.7%
	5	**10.2%**	22.2%
	6	**10.2%**	22.3%
CMUdict-start	4	10.5%	22.8%
	5	10.4%	22.5%
	6	10.3%	22.4%
CMUdict-both	4	10.5%	22.8%
	5	10.4%	22.4%
	6	10.4%	22.4%

Table 5: Results of different language model configurations.

[13] https://huggingface.co/jimregan/psst-partial-timit